ASSIGNMENT 2

NAME: SAILIK PANDEY
ROLL: 002110501132
SEC-A3 BCSE II

1. Two numbers MNH and KLH are stored in 2050H and 2051H, respectively. Write a program to assemble them as NKH and LMH store them in 2052H and 2053H.

Assembler Output

```
1   3A 50 20        lda 2050h; a=[2050h]
2   E6 0f           ani 0fh; a=0N
3   47              mov b, a; b=a
4   3A 51 20        lda 2051h; a=[2051h]
5   E6 f0           ani f0h; a=K0
6   80              add b; a= a+b (KN)
7
8   16 04           mvi d, 04h;
9   0F              back1: rrc; right rotate
10  15              dcr d; d=d-1
11  C2 0e 08        jnz back1
12  32 52 20        sta 2052h; [2052h]=NK
13
14  3A 50 20        lda 2050h; a=[2050h]
15  E6 f0           ani f0h; a=M0
16  47              mov b, a; b=a
17  3A 51 20        lda 2051h; a=[2051h]
18  E6 0f           ani 0fh; a=0L
19  80              add b; a= a+b (ML)
20
21  16 04           mvi d, 04h;
22  0F              back2: rrc; right rotate
23  15              dcr d; d=d-1
24  C2 24 08        jnz back2
25  32 53 20        sta 2053h; [2053h]=NK
26
27  76              hlt
```

2. Two numbers A & B are stored in 2050H and 2051H, respectively. Write a program to perform A×B and store the result in 2052H and 2053H.

Assembler Output

```
1  21 00 00   lxi h, 0000h; HL=0000h
2  11 00 00   lxi d, 0000h; DE=0000h
3  3A 50 20   lda 2050h; A=[2050h]
4  C6 00      adi 00h; checking 1st operand==0
5  CA 1d 08   jz store; 0*something=0, so store 0
6  5F         mov e, a; DE=00XXh
7
```

Assembler Output

```
3A 51 20   lda 2051h; A=[2051h]
C6 00      adi 00h; checking 2nd operand==0
CA 1d 08   jz store; something*0=0, so store 0
4F         mov c, a; C=A

19         back: dad d; HL=HL+DE
0D         dcr c; C=C-1
C2 18 08   jnz back

7D         store:mov a, l; A=L
32 52 20   sta 2052h; [2052h]=lower byte of result
7C         mov a, h; A=H
32 53 20   sta 2053h; [2053h]=higher byte of result

76         hlt
```

3. N numbers are stored in consecutive m/m location starting from 2050H. The value N is stored in 204FH.
   a) Find the maximum among the N numbers.
   b) Find the minimum among the N numbers.
   c) Sort the N numbers in ascending order.

Assembler Output

```
3A 4f 20   lda 204Fh; a = N(count)
3D         dcr a;
47         mov b, a; b = a (outer loop count)

21 50 20   back2: lxi h, 2050h; HL = 2050h
3A 4f 20   lda 204Fh; a = N(count)
3D         dcr a;
4F         mov c, a; c = a (inner loop count)

7E         back1: mov a, m; d = 1st number
23         inx h; HL = 2051
BE         cmp m; a-b
DA 1c 08   jc skip; i.e, a<b
CA 1c 08   jz skip; i.e, a==b
           ;a>b condition
```

```
5E          mov e, m; e = 2nd number
77          mov m, a; [2051]= 1st number
2B          dcx h; HL = 2050
7B          mov a, e; a = 2nd number
77          mov m, a; [2050] = 2nd number
23          inx h; HL = 2051
0D          skip: dcr c; c = c-1
C2 0d 08    jnz back1
05          dcr b; b = b-1
C2 05 08    jnz back2
            ; getting the smallest number
3A 50 20    lda 2050h; a = 1st number(smallest)
32 60 20    sta 2060h; [2060h] = a
            ; getting the largest number
3A 4f 20    lda 204Fh; a = N(count)
3D          dcr a; a = a - 1
21 50 20    lxi h, 2050h; HL = 2050h
85          add l; a = a + l
D2 36 08    jnc store; if no carry
24          inr h; h = h + 1
6F          store:mov l, a; l = a
7E          mov a, m; a = largest number
32 61 20    sta 2061h; [2061h] = last number(largest)
76          hlt;
```

d) Sort the N numbers in descending order.

Assembler Output

```
3A 4f 20    lda 204Fh; a = N(count)
3D          dcr a;
47          mov b, a; b = a (outer loop count)

21 50 20    back2: lxi h, 2050h; HL = 2050h
3A 4f 20    lda 204Fh; a = N(count)
3D          dcr a;
4F          mov c, a; c = a (inner loop count)

7E          back1: mov a, m; d = 1st number
23          inx h; HL = 2051
BE          cmp m; a-b
D2 1c 08    jnc skip; i.e, a>b
CA 1c 08    jz skip; i.e, a==b
            ;a<b condition
```

```
5E          mov e, m; e = 2nd number
77          mov m, a; [2051]= 1st number
2B          dcx h; HL = 2050
7B          mov a, e; a = 2nd number
77          mov m, a; [2050] = 2nd number
23          inx h; HL = 2051
0D          skip: dcr c; c = c-1
C2 0d 08    jnz back1
05          dcr b; b = b-1
C2 05 08    jnz back2
76          hlt;
```

4. N numbers are stored in consecutive m/m location starting from 2050H. The value N is stored in 204FH. Write a program to copy the even and odd numbers starting from 2100H and 2200H, respectively. Store the total no. of even and odd numbers in 2300H and 2201H, respectively.

Assembler Output

```
3A 4f 20    lda 204fh; a = N (count)
06 00       mvi b, 00h;
4F          mov c, a; c = a
21 50 20    lxi h, 2050h; HL = 2050h

            ;even numbers
11 00 21    lxi d, 2100h; DE= 2100h
7E          back1: mov a, m; a = 1st number
0F          rrc; right rotate -> last bit to carry
DA 1d 08    jc skip1; skip if lsb == 1(odd)
04          inr b; b = b + 1;
EB          xchg; HL and DE exchanged values
0F          rrc; rotating 7 times -> original num
0F          rrc
0F          rrc
0F          rrc
0F          rrc
0F          rrc
0F          rrc
77          mov m, a; [2100] = 1st even number
23          inx h; HL = 2101h
EB          xchg; HL and DE exchanged values
23          skip1: inx h;HL = 2051h
0D          dcr c; c = c - 1
C2 0c 08    jnz back1
78          mov a, b; a = count of even
32 00 23    sta 2300H; [2300] = count of even

```

```
            ;odd numbers

3A 4f 20    lda 204fh; a = N (count)
06 00       mvi b, 00h;
4F          mov c, a; c = a
21 50 20    lxi h, 2050h; HL = 2050h

11 00 22    lxi d, 2200h; DE= 2200h
7E          back2: mov a, m; a = 1st number
0F          rrc; right rotate -> last bit to carry
D2 43 08    jnc skip2; skip if lsb == 0(even)
04          inr b; b = b + 1;
EB          xchg; HL and DE exchanged values
```

```
0F          rrc; rotating 7 times -> original num
0F          rrc
0F          rrc
0F          rrc
0F          rrc
0F          rrc
0F          rrc
77          mov m, a; [2200] = 1st odd number
23          inx h; HL = 2201h
EB          xchg; HL and DE exchanged values
23          skip2: inx h;HL = 2051h
0D          dcr c; c = c - 1
C2 32 08    jnz back2
```

```
78          mov a, b; a = count of odd
32 01 23    sta 2301H; [2301] = count of odd
76          hlt

```

5. N numbers are stored in consecutive m/m location starting from 2050H. The value N is stored in 204FH. Write a program to test whether a number stored in 204EH is present in the list. If present, store its position in the list at 204DH; otherwise store FFH.

Assembler Output

```
            ;assuming the number is not present
3E ff       mvi a, FFh; a = FFh
32 4d 20    sta 204Dh; [204dh] = FFh

3A 4f 20    lda 204fh; a = count
4F          mov c, a; c = a(count)
16 01       mvi d, 01h; d = 01h(index)

21 50 20    lxi h, 2050h; HL = 2050h
3A 4e 20    lda 204eh; a = target number
47          mov b, a; d = a(target number)

7E          back: mov a, m; a = 1st number;
90          sub b; a = a - b(target number);
C2 1b 08    jnz skip; jump if result!=0
```

```
7A          mov a, d; a = current index
32 4d 20    sta 204Dh; [204eh] = index result
23          skip: inx h; HL = 2051h;
14          inr d; d = d + 1(index)
0D          dcr c; c = c - 1(count)
C2 12 08    jnz back; if c!=0
76          hlt
```